

# Hello Message Scheme Enhancement in CRMANET

Jingwen Bai\*, Tao Liang<sup>†</sup>, Yan Sun\*, and Chris Phillips\*

\*School of Electrical Engineering and Computer Science, Queen Mary University of London

Email: {j.bai, yan.sun, chris.i.phillips}@qmul.ac.uk

<sup>†</sup>School of Electrical Engineering and Computer Science, Queen Mary University of London

Email: t.liang@se11.qmul.ac.uk

**Abstract**—All these years, a lot of efforts have been put upon how to reduce the broadcast overhead consumption in Cognitive Radio enabled Mobile Ad hoc Network (CRMANET). In this work, we propose an improved hello message scheme named Adaptive Classified Hello Scheme (ACHS) adopting classification method in CRMANET. Different from fixed hello interval and content form in frequently used Periodic Hello Message Scheme (PHMS), ACHS categorizes nodes into different classes based on node mobility. Each class in ACHS will be configured with different hello intervals and content format. Given each nodes real-time function performing in CRMANET (on the route or off the route), ACHS employs different strategies. For instance, when nodes are performing data transmission, instead of sending dedicate hello messages, their hello information will be attached into the data message to further reduce the control overhead. Compared with Periodic Hello Message Scheme (PHMS) and Reactive Hello Protocol (RHMS) in simulation, ACHS has improved hello efficiency around 50%.

## I. INTRODUCTION

Due to nodes mobility in CRMANET, how to find a stable route to efficiently transmit data from source to destination absorbs a lot of interests [1] [2] [3] [4]. Discovering and maintaining neighbors via broadcasting hello message has been regarded as an important operation in order to achieve a good performance in route discovery, route reconstruction and route maintenance. However given the limitation of device battery, improving broadcasting efficiency becomes a key point to save transmission energy and improve the performance of whole network. PHMS is one of the traditional methods to support neighbor table updates and maintenance. Specifically, nodes periodically advertise hello messages to their neighbors indicating their existence. However, fixed interval is not suitable for real network circumstances. For example, the hello message with short periodic interval may cause unnecessary congestion, while long periodic interval may cause slow reflection of the network changes. According to the routing protocols in [5] [6] [7] [8], neighbour tables are used to verify whether one of its neighbours is the next hop on an active route. In this case, the neighbour table helps with route establishment and link broken detection. For other cases in [5] [8], nodes only depend on the neighbour table to forward data, which means that except the link detection, neighbor table also

contributes towards routing function. Therefore, in order to meet the high performance requirements of neighbour table for certain routing protocols, a short broadcasting interval for hello message is preferable. However, short interval can also lead to resources exhausting quickly, like battery [9], sometimes even causing data congestion. Based on the above issues, the goal of this work is to propose a brand new hello broadcast scheme called ACHS which provides an efficient neighbour table updates and guarantees the overall network performance at the same time.

The rest of the paper is organized as follow: Section II introduces some related works. Section III gives the details of ACHS design. In Section IV, the simulation results and analysis of ACHS are described then conclusions are presented in Section V.

## II. RELATED WORK

In order to improve the efficiency of hello message broadcasting in CRMANET, certain schemes have been proposed already, mainly in two aspects: periodic broadcasting and reactive broadcasting. PHMS [10] is the most traditional periodic broadcasting scheme which has been widely adopted for the neighbour table updates and maintenance in CRMANET. When a node receives a hello message from its neighbours, the node will create a new entry or update the corresponding entry in its neighbour table. Within a pre-defined period of time, if the node does not get any hello message from the same neighbour, the entry in the neighbor table will be marked as expired and deleted. Only the neighbour entry in the neighbour table is active, the corresponding neighbor node can contribute to establishing a valid route. PHMS became popular due to its implementation simplicity. However, fixed interval does not always work for all scenarios. The short periodic interval may cause network congestion and extra resource consumption, while long periodic interval might lead to delayed reflection of network topology changes. The design idea of RHMS is drawn by on-demand routing protocols such as AODV [7] and ABR [11]. In RHMS, the nodes including source node and intermediate node only build up the neighbor table when then need it. For the source node, once the data from application layer comes to IP layer, it will be buffered before the connectivity establishment procedure finished in

the neighbor table. The procedure is like this: First, a hello request packet will be broadcasted by the request node. Within the RESP\_WAIT\_TIME period (RESP\_WAIT\_TIME period: a maximum time to wait for a valid hello response), if no hello response is received, the hello request packet will be resended and the maximum attempts is set as the parameter MAX\_RETRIES [10]. As long as the neighbour table is set up, the source node will send the route request packet to discovery a route. After the route is built, the data in the buffer will be sent out to the next hop. For the intermediate nodes, if the hello packets from the other nodes are received, they will be triggered to broadcast hello messages. Because of the node mobility, neighbour table will be deleted according to the setting of parameter NBR\_VALID\_TIME (NBR\_VALID\_TIME: a maximum time to keep the neighbor table valid before the next hello is received). However, buffering packets can lead to long end-to-end delay.

Given all of above studies, ACHS improves the hello performance mainly from two aspects. First ACHS dynamically controls the hello interval to diminish the number of the hello message. Second ACHS re-designs the content and the size of the hello message. ACHS proposed in this paper adopts a novel classification method to adjust nodes into different classes according to node state (on the route or off the route). According to the node mobility, different hello interval and content will be set to each class in ACHS. What is more, during data transmission, the part of hello information will be added into the data packet, and meanwhile the node will cancel sending the next hello message. This design will reduce the unnecessary hello message without damaging the link status detection.

### III. DESIGN

#### A. Structure Overview

ACHS uses classification method to categorize nodes into different classes. According to the different classes, different hello intervals will be assigned and the payload of data packets will be modified. Specifically, if nodes are sending data, the content of hello message can be embedded into the data packet to maintain the neighbor relationship among the nodes on the route. And the nodes around the route can also use the Cognitive Radio feature to extract the hello message in the data to maintain their neighbor information. This design intends to reduce unnecessary hello messages and guarantee sufficient the link detection simultaneously. The classification strategy is: those nodes on the route are defined as Class 1 and the nodes off the route belong to Class 2. However, in some protocols, some nodes have special functions and play important role to improve the performance, such as the nodes in Cooperative Table in CRCPR [12]. Therefore, considering expansibility, Class 1 can be further separated into two sub-classes: Class 1A and Class 1B, to allow this design to be easily expanded for special demands. Considering the hello message structure in ACHS, we design two fields in the traditional hello header format making use of the reserved bits to allow compatibility: INTERVAL and VELOCITY. The modified packets formats

are illustrated in TABLE I and TABLE II. TABLE I shows the new hello message called Simple Hello Format. The protocols, whose hello message does not include neighbour IP address such as AODV and ABR can use this format..

TABLE I  
SIMPLE HELLO FORMAT

Original Headers	Interval	Velocity
------------------	----------	----------

TABLE II shows Rich Hello Format. The hello message in the protocol, such as OSPF and CRCPR, who contains neighbour IP address will apply this format.

TABLE II  
RICH HELLO FORMAT

Original Headers	Interval	Velocity	neighbour Address [0]	...	neighbour Address [n]
------------------	----------	----------	-----------------------	-----	-----------------------

#### B. Class 1: Nodes on the route

All the nodes belonging to Class 1 are on the route, which means they will forward data from source node towards the destination node. The basic principle for Class 1 is called "Hello Embed", which means hello message contents can be embedded into data packet to reduce the regular hello transmission. However, the embedding interval will be set dynamically. After the hello content is embedded, the original data part can only be retrieved by the valid next hop on the route. In the fact, the data packet can be received by all neighbours based on CR feature in wireless transmission. When non-next hop neighbours hear the transmission, they will only extract the hello contents to update their neighbour information, and then destroy the packet. In the following TABLE III, three different time intervals for hello message are set according to different time range.  $t_d$  is defined as the time when the data with hello content has been sent successfully and at this moment the next regular hello message is canceled which should have been sent at time  $t_2$ . A new timer for the next hello message will be set via comparing the threshold time  $T_{th}$  and  $|t_2 - t_d|$  in different time range.

TABLE III  
PARAMETER FOR HELLO EMBED

Time Range	Time Interval from $t_d$ to next hello
$T_{th1} > t_2 - t_d > 0$	Tchange1
$T_{th2} > t_2 - t_d > T_{th1}$	Tchange2
$t_2 - t_d > 0T_{th2}$	Tchange3

For one time range, there must be two time thresholds: lower one and higher one. The lower threshold will be seen as  $B \times \text{Hello\_Interval}$  where  $B$  is a coefficient. (For example, if time range is  $T_{th1} > t_2 - t_d > 0$ , 0 as the lower time threshold will be equal to  $B \times \text{Hello\_Interval}$ , then  $B = 0$ ). And the value of Tchange is equal to  $C \times \text{Hello\_Interval}$  where  $C$  is another coefficient. (For example, if time range is

$T_{th1} > t_2 - t_d > 0$ , then  $Tchange1 = C \times Hello\_Interval$ ). Therefore, once the coefficient  $C$  is decided, the value of  $Tchange$  will be found. Therefore, in the premise that the new timer for next hello message can still maintain the neighbour relationship effectively, the Equation (1) can be gained to show the relationship between  $B$  and  $C$ .

$$\begin{aligned} &Hello\_Interval - B \\ &\times Hello\_Interval + C \\ &\times Hello\_Interval \leq Allow\_Hello\_Loss \\ &\times Hello\_Interval \end{aligned} \quad (1)$$

Then, we can modified Equation (1) and obtain the Equation (2) which can be used to calculate coefficient  $C$  once  $Allow\_Hello\_Loss$  and  $B$  has be defined.

$$C \leq Allow\_Hello\_Loss - 1 + B \quad (2)$$

Class 1 contains two sub-classes: Class 1A and Class 1B. Both of these two sub-classes will inherit the Hello Embed Scheme, but more specific processes in terms of the hello message content and interval will be re-designed respectively.

1) *Nodes without Special Functions*: The nodes on the route without special functions will be regarded as Class 1A for hello message management.

If the protocol only applies Simple Hello Format for the hello message, it will not change the hello message format at any time; if the Rich Hello Format is applied in the protocol, the choice of different hello message formats will obey this principle: after a node has sent two hello messages with Simple Hello Format and its neighbour information (a new neighbour joining or an old neighbour leaving) changes at this moment, the node will broadcast the hello message with Rich Hello Format for the next interval. Otherwise, the node broadcasts the hello with Simple Hello Format.

Although these nodes are not with special functions, they can influence data transmission as they are on the route which leads to a smaller hello interval needed for these nodes to maintain the route effectively. Therefore, we propose a interval decision scheme based on the node velocity. The velocity can be got according to the formula below:

$$V = \frac{\sqrt{(X_{new} - X_{old})^2 + (Y_{new} - Y_{old})^2}}{t_{new} - t_{old}} \quad (3)$$

In Equation 3,  $t_{old}$  and  $t_{new}$  are the old observation time and new observation time respectively.  $X_{new}$  and  $Y_{new}$  are X position and Y position of a node at the time  $t_{new}$ .  $X_{old}$  and  $Y_{old}$  are X position and Y position of a node at the time  $t_{old}$ . Then the velocity  $V$  can be calculated accordingly.

In order to get more specific hello intervals, the nodes in Class 1A can be further classified into three subclasses: Class 1A-a, Class 1A-b and Class 1A-c. The subclasses are decided by the comparison result between Determination Velocity ( $V_{de}$ ) and Threshold Velocity ( $V_{th}$ ) which has been shown in TABLE IV.  $V_{th}$  is a scenario-selectable value. Because there are three sub-classes, two different  $V_{th}$  will be defined.

$V_{de}$  can be used to decide which sub-class a node belongs to via comparing with  $V_{th}$ . The value of  $V_{de}$  will be calculated according to the following principle: A node will detect all its neighbours' velocities by CR and choose one velocity with maximum value to observe. If the value of chosen velocity changes to a larger value three times continuously or changes to a smaller value three times continuously, then the average of these three new speeds will be used as the determination velocity  $V_{de}$ . During the period of observation, if the observed node with maximum velocity changes, a new observation will be set.

An example in Figure 1 is used to illustrate the above process. Before sending the hello message, node B observes that the velocity of node A has the maximum value ( $V_1$ ) among its neighbours. So  $V_1$  will be regarded as the comparable velocity  $V_c$ . With the similar definition,  $V_2, V_3, \dots, V_n$  is the velocity observed before the *second, third, ..., n<sub>th</sub>* hello is sent out. Thus, the  $V_m$  is the observed velocity of the node before the *m<sub>th</sub>* hello is sent out. If  $V_{m-1} > V_c$ ,  $V_m > V_c$  and  $V_{m+1} > V_c$  or  $V_{m-1} < V_c$ ,  $V_m < V_c$  and  $V_{m+1} < V_c$ , then node B can get its own Determination Velocity:  $V_{de} = (V_{m-1} + V_m + V_{m+1}) / 3$ . Meanwhile,  $V_{m+1}$  will be regarded as a new comparable velocity and the next observation starts.

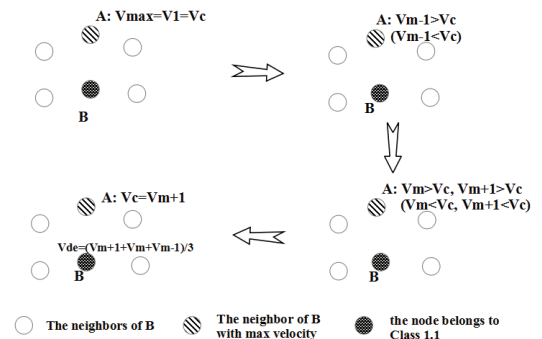


Fig. 1. Setting Determination Velocity

TABLE IV  
PARAMETERS FOR CLASS 1A

Velocity Range	Subclass	Hello Interval
$V_{th1} > V_{de}$	1A-a	h1A-a
$V_{th2} > V_{de} > V_{th1}$	1A-b	h1A-b
$V_{de} > V_{th2}$	1A-c	h1A-c

2) *Nodes with Special Functions*: Once the nodes are on the route and also have special functions, they will be regarded as Class 1B. This class can be extended for any routing protocol which involves different hello functions. In this paper, we use CRCPR as an example. The format of hello message in Class 1B follows this principle: as long as the neighbour information of one node changes (a new neighbour joining or an old neighbour leaving), the node broadcasts will broadcast hello message with Rich Hello Format. Otherwise, a hello

with Simple Hello Format will be sent. In CRCPR, there are four nodes in the specific cooperative topology and they can build a Cooperative Table via hello broadcast information. Therefore, the neighbour information for these special nodes plays an important role and a smaller hello interval should be set. Like Class 1A, the nodes in Class 1B can also be classified into subclasses to set more specific hello interval. If one of the four nodes in cooperative topology gets the velocity of the other three nodes by CR, it will calculate the average value as the Determination Velocity ( $V_{av}$ ). Comparing the ( $V_{av}$ ) and Threshold Velocity ( $V_{th'}$ ), the subclass of a node can be decided in TABLE V and hello interval can be confirmed. Afterwards, all these four nodes in the Cooperative Topology will use the same hello interval. As the mobilities of the nodes in different cooperative topology are not the same, the corresponding hello interval will be decided by different cooperative topology. However, if one node occupies two different cooperative topologies, then the hello interval for both of the cooperative topologies will be calculated and the smaller one will be chosen as the hello interval for the nodes in these two cooperative topologies.

TABLE V  
PARAMETERS FOR CLASS 1B

Velocity Range	Subclass	Hello Interval
$V_{th1'} > V_{av}$	1B-a	h1B-a
$V_{th2'} > V_{av} > V_{th1'}$	1B-b	h1B-b
$V_{av} > V_{th2'}$	1B-c	h1B-c

The example is shown in Figure 2. Node A, B, C and D set up a cooperative topology. Node A has the velocity  $V_1$  and it can get the velocities of node B ( $V_3$ ) and D ( $V_4$ ) by CR. Node A can also know the velocity of node C ( $V_2$ ) via the unicast hello relayed from node B or D. Node B, C and D can get to know the velocities of each other by the same method. The average velocity in cooperative topology can be calculated:  $V_{av} = (V_1 + V_2 + V_3 + V_4) / 4$ . Finally, we can decide the hello interval for these four nodes in the cooperative topology.

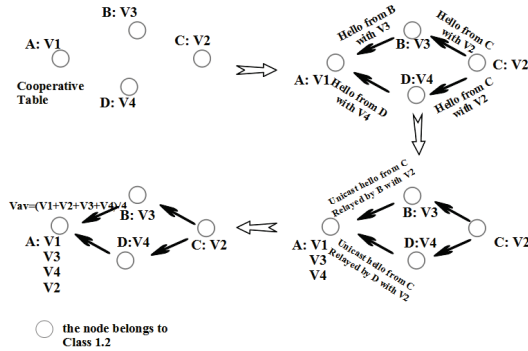


Fig. 2. Setting Determination Velocity

#### C. Class 2: Nodes off the route

All the nodes that are not on the route are classified into Class 2. Since the movement of the nodes in this class rarely

influences the route status, they have the lowest priority. If the Rich Hello Format is applied for the hello message in one protocol, the changes of hello message format will accord this principle: after a node has sent four hello messages with Simple Hello Format and its neighbour information (a new neighbour joining or an old neighbour leaving) changes at this moment, the node will broadcast the hello message with Rich Hello Format for next interval. Otherwise, a hello message with Simple Hello Format will be sent. The Hello Interval  $h_2$  will be set for the nodes in this class and it is longer than the interval in any other classes. This principle can reduce the average number of hello message and save transmission energy as well.

### IV. SIMULATION ANALYSIS

#### A. Stable scenarios

Figure 3 shows the scenario with 15 fixed nodes. AODV and CRCPR are simulated in this scenario to route data from N\_1 to N\_2. For AODV, three hello mechanisms are tested: PHMS, RHMS and the new ACHS. For CRCPR, PHMS and the new ACHS are simulated. The transmission range of each node is 75m. Red arrows show the established route by AODV, and blue arrows indicate the setup route by CRCPR. Some other parameters are shown in Table IV. These parameters are set just as suggested in AODV [7] and CRCPR [12].

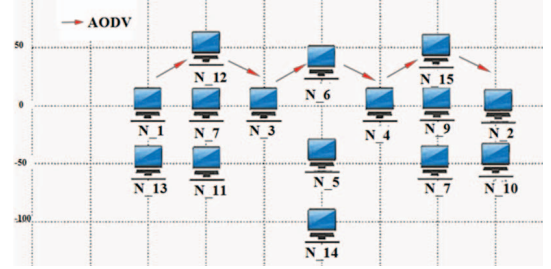


Fig. 3. Stable Scenario

TABLE VI  
PARAMETERS FOR STABLE SCENARIOS

Date Interval	1s
PHMS Hello Interval	1s
RHMS Hello Interval	1s
ALLOW HELLO LOSS	2

1) *Results under AODV*: The number of nodes in the scenario increases from 20, 25 to 30 to analysis the end-to-end delay, hello efficiency and the hello message rate. Hello efficiency is defined to evaluate the network performance considering the broadcasting hello message and the throughput. The definition is shown below:

$$Hello\_efficiency = \frac{Throughput}{Hello\_message\_rate} \quad (4)$$

In Equation 4, we can see the meaning of hello efficiency is how many data packets can be transmitted successfully based



on each hello message which can be used to evaluate the hello message broadcasting efficiency.

Figure 4 shows the End-to-End Delay. The performance of the ACHS is almost the same with PHMS and obviously lower than the RHMS. The reason is that in RHMS, nodes only start broadcasting hello messages when they need to send data so that it will take more time to build the neighbour table before discovering the route. Therefore, ACHS and PHMS have a better performance than RHMS in the terms of End-to-End Delay.

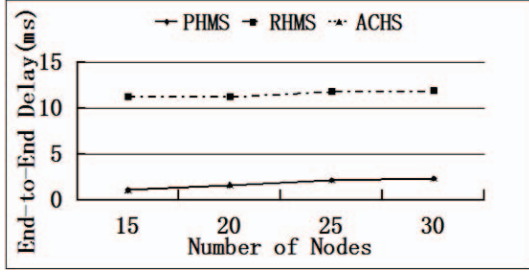


Fig. 4. End-to-End Delay in AODV

In Figure 5, as the number of nodes increases, the trend of hello efficiency is decreasing. This is due to the reason that source node keeps the same data transmission speed but the number of broadcasting hello messages grows. Since RHMS can reduce the hello message transmission before data transmission, its curve is lower than the PHMS. And for ACHS, it embeds the content of hello message into data, the number of transmitting hello messages is lowest among these three methods. Therefore, compared with PHMS and RHMS, ACHS can reduce the number of hello messages without harming the data throughput.

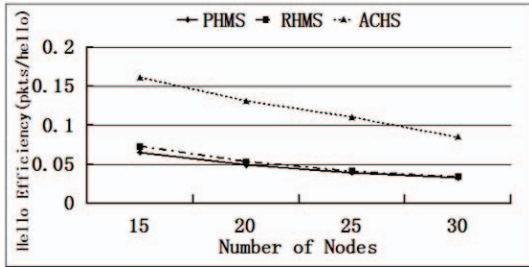


Fig. 5. Hello Efficiency in AODV

2) *Results under CRCPR*: Figure 6 indicates that the rate of hello messages has been reduced greatly by adopting ACHS. Thus, in CRCPR, ACHS has a better performance than PHMS.

### B. Mobile scenarios

Figure 7 shows a scenario with 50 nodes. The transmission range of each node is 75m. AODV and CRCPR are simulated to route data from N\_1 to N\_2. All other nodes are fixed except the nodes with the green arrow. The moving nodes are configured following uniformly accelerated motion with

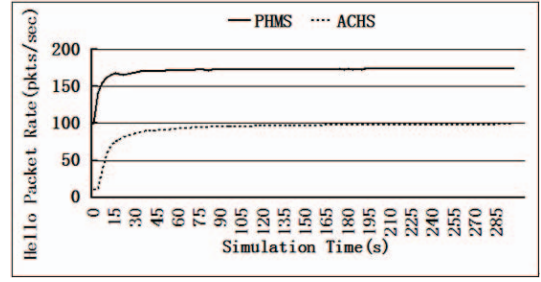


Fig. 6. Hello Message Rate in CRCPR

accelerated velocity  $0.1m/s^2$ . The red arrows and pink arrows are the routes established in CRCPR and AODV respectively. The blue arrows indicate reconstructed route for both AODV and CRCPR after link break. The other parameters are set the same with stable scenario. The velocity of the moving nodes will be increased from 0.1m/s, 0.3m/s to 0.5m/s to analysis the performance of end-to-end delay, hello efficiency and the hello message rate. For AODV, three hello mechanisms are tested: PHMS, RHMS and the new ACHS. For CRCPR, PHMS and the new ACHS are simulated. In order to do the classification for ACHS scheme, the thresholds are set in Table VII:

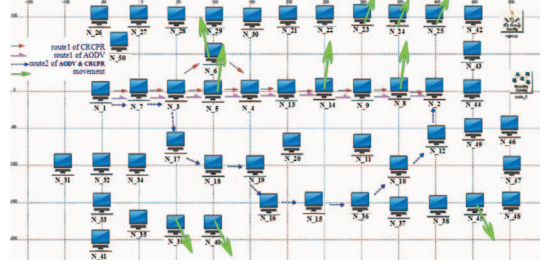


Fig. 7. Moving Scenario

TABLE VII  
THRESHOLDS FOR ACHS SCHEME

$V_{th1}$	0.3	$V_{th1'}$	0.2
$V_{th2}$	0.8	$V_{th2'}$	0.7
$h1A - a$	1.5	$h1B - a$	1.5
$h1A - b$	1.35	$h1B - b$	1.2
$h1A - c$	1.1	$h1B - c$	0.9

1) *Results under AODV*: Figure 8 indicates the End-to-End Delay in AODV. The results seem similar with the stable scenario. The End-to-End Delay of RHMS is the highest because of its on-demand design. So ACHS and PHMS have a better performance than RHMS.

Figure 9 illustrates the hello efficiency. ACHS achieves the best performance. The trend of hello efficiency of three different schemes stay stable in this moving scenario because the number of nodes in this scenario keeps unchanged during simulation as well as the number of broadcasting hello message. The reason why PHMS has the worst performance is that the number of hello messages in on-demand scheme is

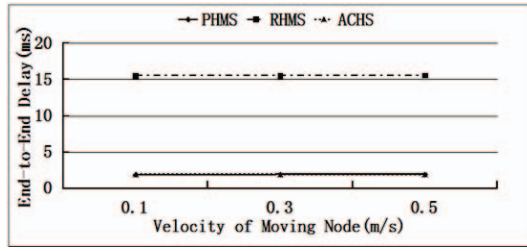


Fig. 8. End-to-End Delay in AODV

decreasing as more nodes moves out of transmission range. For RHMS, although the decreasing speed becomes slower with the incensement of moving node, the number of hello message is higher than ACHS. Therefore, the ACHS shows an outstanding performance.

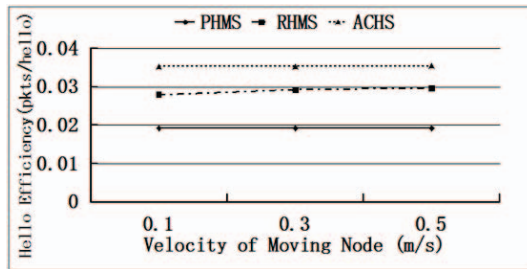


Fig. 9. Hello Efficiency in AODV

2) *Results under CRCPR*: Figure 10 indicates the results are the same as the stable scenario. The transmitting rate of hello message has been diminished a lot in ACHS.

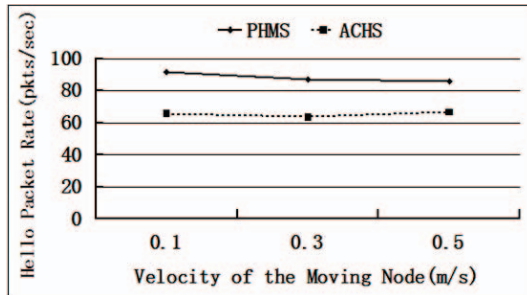


Fig. 10. Hello Message Rate in CRCPR

## V. CONCLUSION

This paper proposes Adaptive Classified Hello Scheme (ACHS) for the routing protocols designed for CRMANET. It aims at improving hello broadcasting efficiency especially to reduce the congestion and save energy. The design uses classification method to adjust nodes with suitable hello intervals. What is more, in order to reduce the unnecessary hello message and meanwhile satisfy the link detection, the content of hello message will be embedded into the data packet during data transmission. The simulation results in both stable scenarios and mobile scenarios under AODV and CRCPR

protocols prove that compared with PHMS and RHMS, ACHS has eliminated the number of hello message dramatically and guarantee the throughput at the same time. Meanwhile, the end-to end delay has been guaranteed. The last but not least, ACHS is an easy-deployed and extensible scheme which means it is not only easy to be set up into routing protocols with hello scheme in CRMANET to help improve broadcasting efficiency, but also able to be updated to adapt different scenarios via exploring more classification methods.

## REFERENCES

- [1] P. Tomer and M. Chandra, "An application of routing protocols for vehicular ad-hoc networks," in *Networking and Information Technology (ICNIT), 2010 International Conference on*, June 2010, pp. 157–160.
- [2] B. Sathyaraj and R. Doss, "Route maintenance using mobility prediction for mobile ad hoc networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, Nov 2005, pp. 6 pp.–101.
- [3] B. Xue, P. Ren, and S. Yan, "Link optimization ad-hoc on-demand multipath distance vector routing for mobile ad-hoc networks," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, Sept 2009, pp. 1–6.
- [4] S. Kumar, R. Rathy, and D. Pandey, "Traffic pattern based performance comparison of two reactive routing protocols for ad hoc networks using ns2," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, Aug 2009, pp. 369–373.
- [5] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (dream)," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '98. New York, NY, USA: ACM, 1998, pp. 76–84. [Online]. Available: <http://doi.acm.org/10.1145/288235.288254>
- [6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, Oct. 1994. [Online]. Available: <http://doi.acm.org/10.1145/190809.190336>
- [7] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, Feb 1999, pp. 90–100. [Online]. Available: <http://moment.cs.ucsb.edu/AODV/ID/draft-ietf-manet-aodv-13.txt>
- [8] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254. [Online]. Available: <http://doi.acm.org/10.1145/345910.345953>
- [9] C. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Pearson Education, 2001. [Online]. Available: <https://books.google.co.uk/books?id=9fUchlgTXDQC>
- [10] V. Giruka and M. Singhal, "Hello protocols for ad-hoc networks: overhead and accuracy tradeoffs," in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, June 2005, pp. 354–361.
- [11] D. JOHNSON, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," *IETF-Request-for-Comments, rfc4728. txt*, 2007.
- [12] J. Bai, Y. Sun, and C. Phillips, "Crcpr: Cognitive radio cooperative communication routing protocol for mobile ad hoc networks," in *Telecommunications (ICT), 2016 23rd International Conference*, May 2016.